《编译原理》课程设计总结

学号：1553534

姓名：李帅

专业：计算机科学技术

**语法分析器设计**

**1.1题目**

1．根据算符优先分析算法，编写一个语法分析程序，可以选择以下三项之一作为分析算法中的输入：

（1）直接输入根据已知文法人工构造的算法优先关系矩阵。

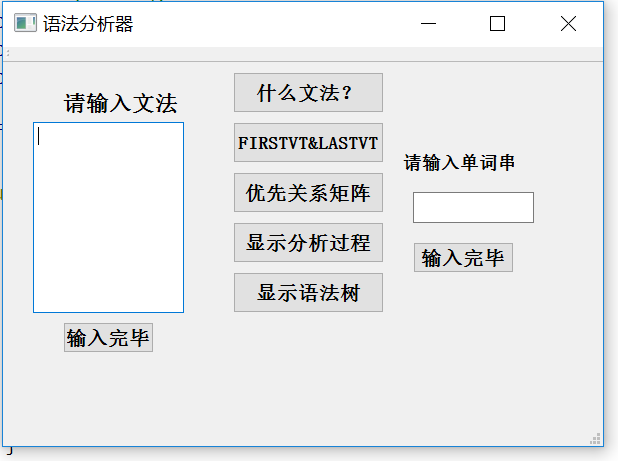
（2）输入已知文法和FIRST、LAST集合，由程序自动生成该文法的算法优先关系矩阵。

（3）输入已知文法，由程序自动生成该文法的算法优先关系矩阵。

2．程序具有通用性，即所编制的语法分析程序能够适用于不同文法以及各种输入单词串，并能判断该文法是否为算符文法和算符优先文法。

3．对输入的一个文法和一个单词串，程序能正确判断此单词串是否为该文法的句子，并要求输出分析过程和语法树。

下图为主界面：

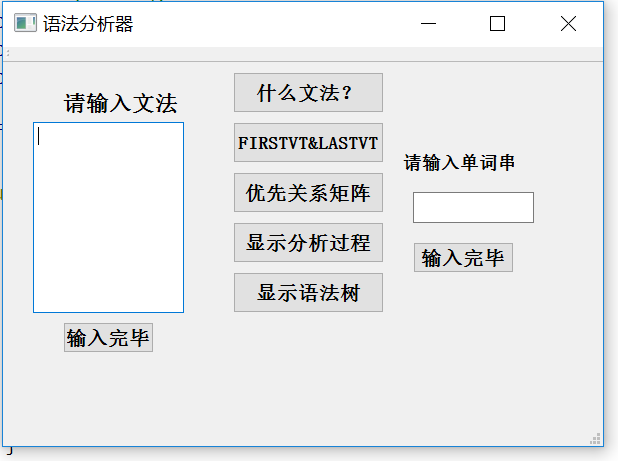


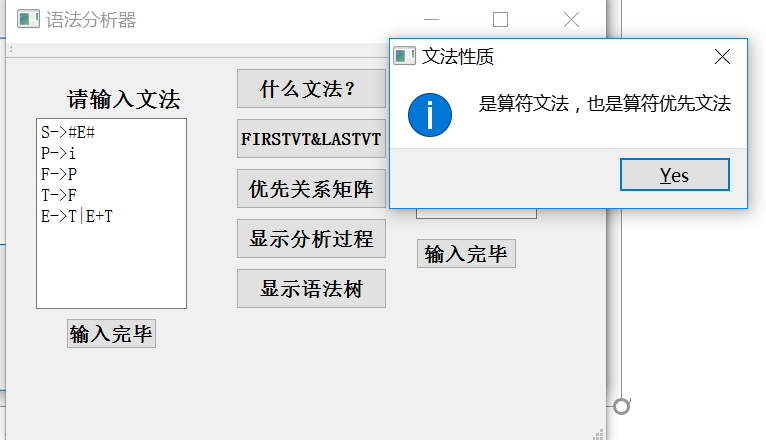
**1.2软件功能**

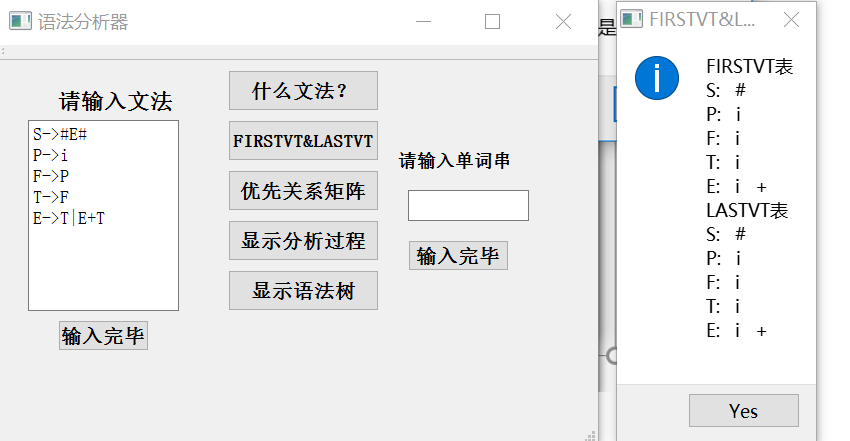
根据题目要求，**本软件应实现以下功能**：

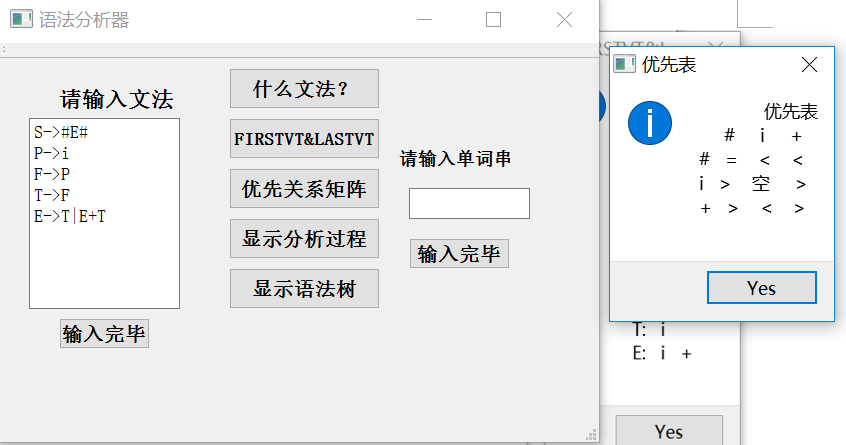
1. 任意输入文法，点击输入完毕
2. 显示输入文法是不是算符文法，以及是不是算符优先文法
3. 自动分析FIRSTVT和LASTVT，并显示出表格
4. 自动分析输出优先表
5. 任意输入单词串，点击确认输入后，如果输入错误，会报错（后续操作不用执行）
6. 点击按钮，显示出移进归约过程
7. 画出语法树

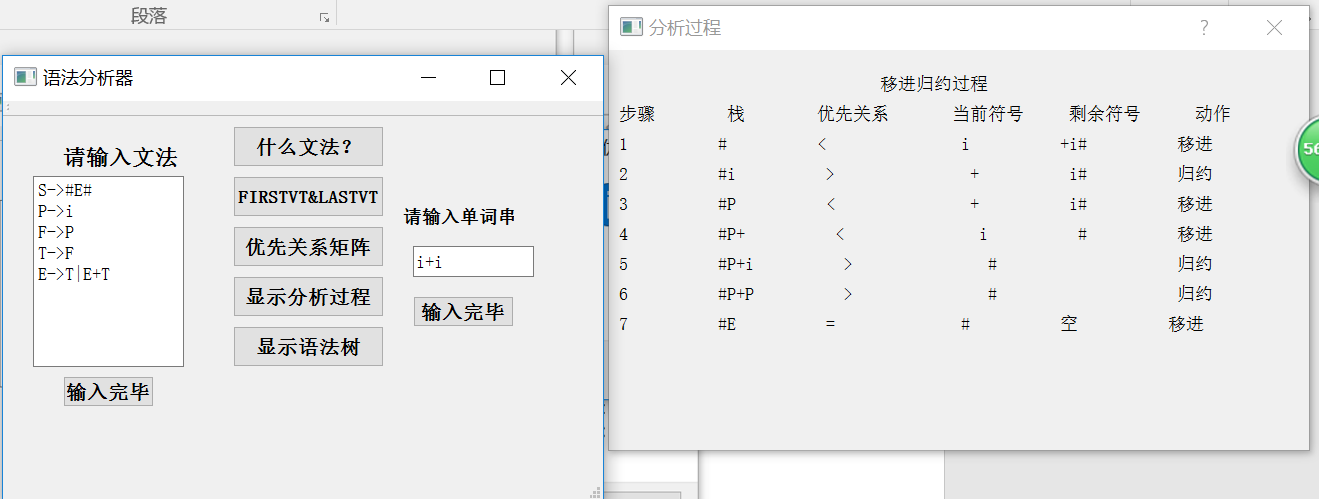
如下图：

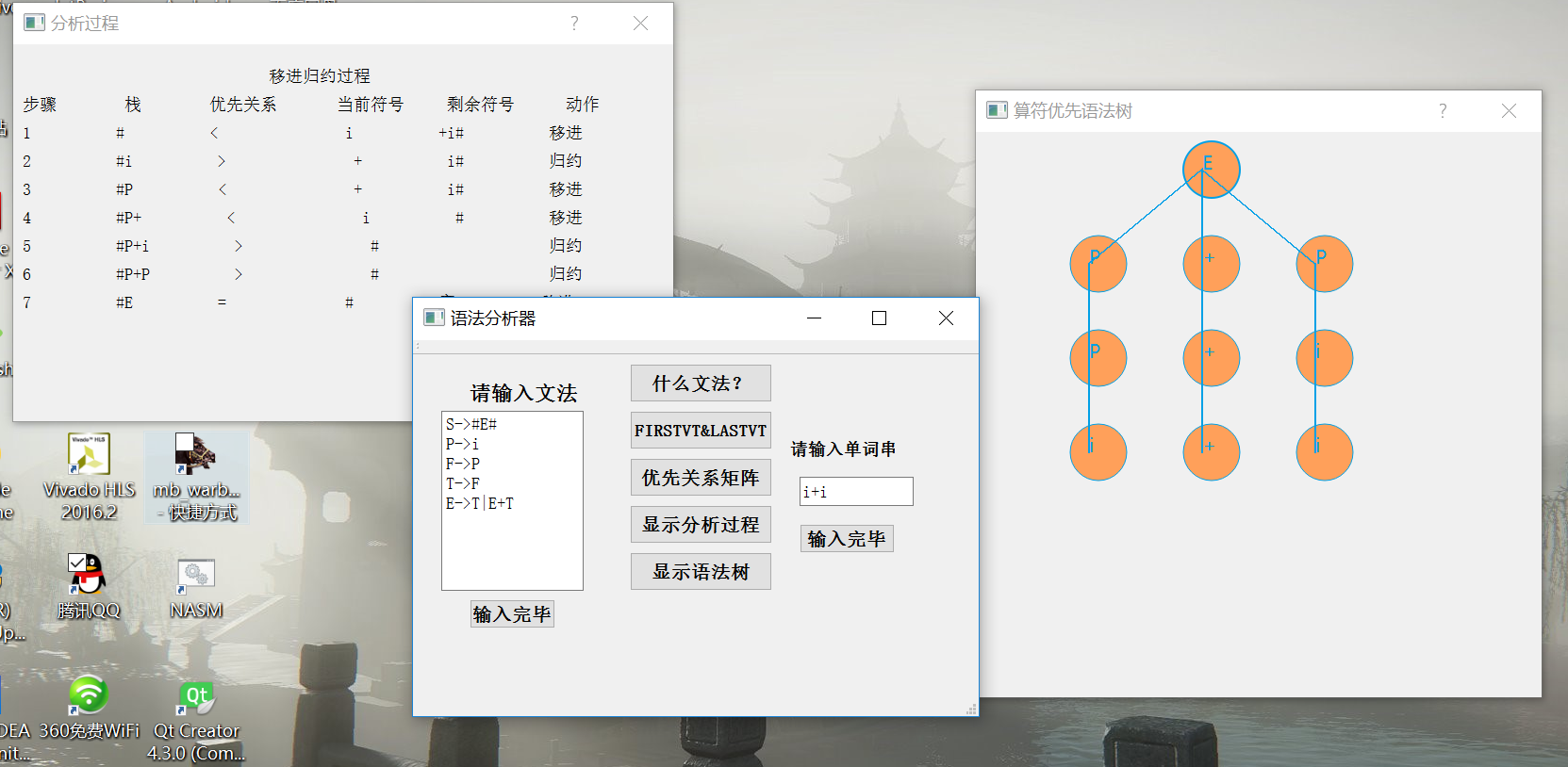












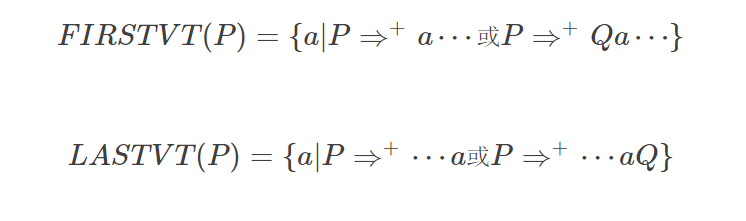
**实现方式：**

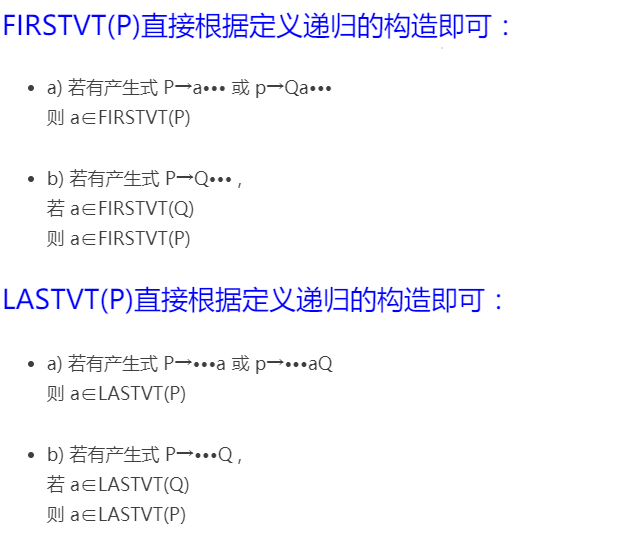
1. 使用Qt的QPaintEvent，绘画事件机制，画出相应树
2. 使用LineEdit控件，以及PushButton控件，以及深度优先算法，递归得出FIRSTVT和LASTVT。
3. 使用LineEdit控件，以及PushButton控件，以及构造优先表算法，输出优先表。
4. 使用算符优先分析算法，得出分析过程，记录路径
5. 使用记录的路径，画出相应的语法树。

**1.3设计思想**

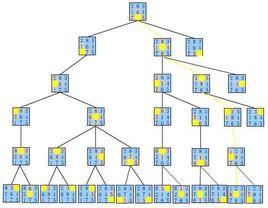
本软件的设计思路就是首先创建一个 mainwindow，使用 PushButton 控件Label控件，TextEdit控件，使用信号槽，当单击相应的按钮时，转到新的继承自QDialog类的窗口，显示相应内容或者显示查询界面，再通过QMessageBox类显示出information,当输入单词串时，会提示报错，故本软件具有一定的容错性能以及纠错提示功能。

**算法设计以及流程说明：**

1. 判断文法性质：在输入时就扫描文法，如果文法的任意产生式的右部含有两个相继的非终结符，那么就不是算符文法，否则就是算符文法，这这一步分析基础上，在构造算符优先表的时候，如果出现算符文法中任何终结符对满足（><=）三种的两种以上就不是算符优先文法，否则就是算符优先文法。
2. FIRSTVT&&LASTVT构造算法：



则很明显，通过深度优先搜索算法，递归得到FIRSTVT和LASTVT



1. 构造优先表算法：按照书上的算法

For每条产生式P－〉X1X2…Xn DO

For i=1;to n-1 Do

Begin

If xi和xi+1 均为终结符　then 置　xi=xi+1

If i<=n-2 且　xi 和　xi+2都为终结符

但Xi+1为非终结 then　置　xi=xi+1

If xi为终结符而xi+1为非终结符 then

For FirstVt(xi+1)中的每个a　DO

置　xi<a;

If xi为非终结符而xi+1为终结符 then

For LastVt(xi)中的每个a　DO

置　a>xi;

END

1. 算符优先分析算法：

K=1; s[k]=’#’

Repeat

把下一个输入符号读进a中；

Ifs[k]属于Vt，then j=k else j=k-1;

While s[j]> a do

Begin

Reapeat

Q=s[j];

If s[j-1]属于Vt then j=j-1 else j=j-2

Until s[j]<Q

把s[j+1]…s[k]归约某个N；

K=j+1

S[k]=N

End of while

If s[j]<a or s[j]=a then

Begin k=k+1; s[k]=a end

Else 出错

UNTIL a=’#’

**1.4逻辑结构和物理结构**

逻辑结构：图形结构，线性结构，树形结构，栈结构，

R={R1,R2}

R1={<mainwindow , window(i)>| 1<=i<=6}

R2={<window(i),data(i).ui(i)>|1<=i<=6}

物理结构：链栈，数组，若干自定义结构体，图，红黑树，动态数组

**自定义结构体结构：**

QMap<node,QChar> ans;//优先关系表

QVector<QChar> VT;//算符

extern QVector<wenfa> VN\_input;//输入的几行文法

QMap<QChar,int> VN\_map;//记录左端大写字符的位置,从1

QSet<QChar> FIRSTVT[Max];

QSet<QChar> LASTVT[Max];

QMap<QChar,int>used;

int vis[Max];//dfs

QVector<QString> hang;//记录每一行的文法

bool flag1=true;//判断是不是算符文法

bool flag2=true;//判断是不是算符优先文法

//建立优先表的节点

struct node{

QChar x;

QChar y;

bool operator <(const node &a)const

{

if(x!=a.x)

return x<a.x;

return y<a.y;

}

};

//存储一行文法语句

class wenfa

{

public:

QChar left;

QVector<QString> right;

wenfa(const QChar& str)

{

left=str;

}

wenfa()

{

}

void insert(QString st)

{

right.push\_back(st);

}

};

**主窗口结构：**

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

explicit MainWindow(QWidget \*parent = 0);

~MainWindow();

private slots:

void on\_pushButton\_clicked();//判断是不是算符优先文法

void on\_pushButton\_2\_clicked();//显示优先表

void on\_pushButton\_6\_clicked();//确认输入键

void on\_pushButton\_7\_clicked();//显示FIRSTVT && LASTVT

void on\_pushButton\_3\_clicked();//确定输入字符串

void on\_pushButton\_4\_clicked();//显示分析过程

void on\_pushButton\_5\_clicked();//显示树

private:

Ui::MainWindow \*ui;

Dialog1 w;

Dialog2 u;

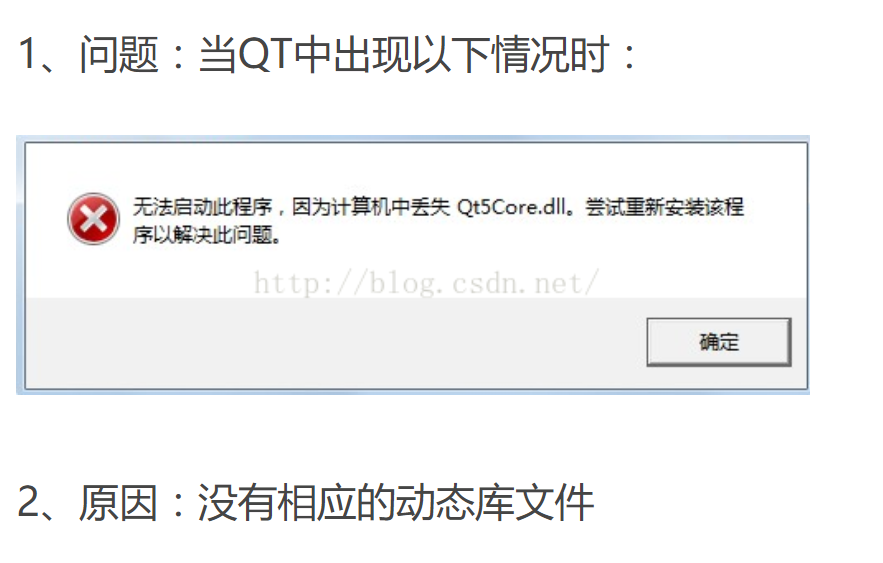
};

#endif // MAINWINDOW\_H

**1.5开发平台**

本软件采用的开发平台是Windows Qt5.9社区版。

注意：**本实验exe文件只有在Qt环境中运行，如果在没安装Qt环境中运行时，需要配置dll文件，按照如下博客配置动态库文件。**



**部分电脑报错解决方案：**

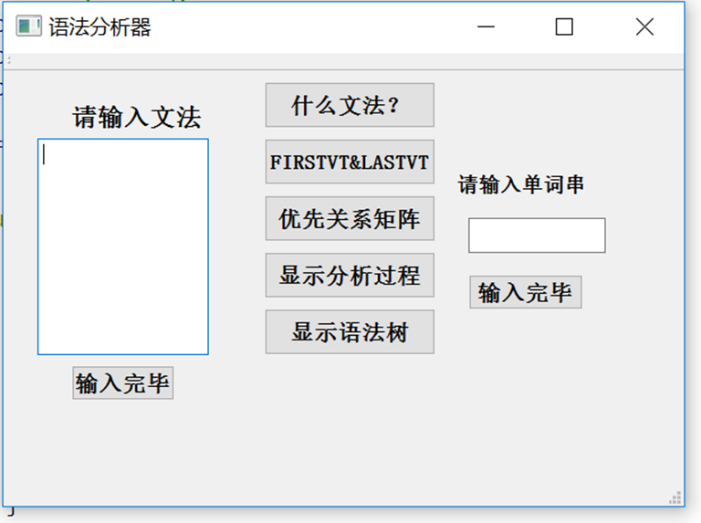
**http://blog.csdn.net/wxf2012301351/article/details/52182769**

**1.6系统的运行操作分析说明**

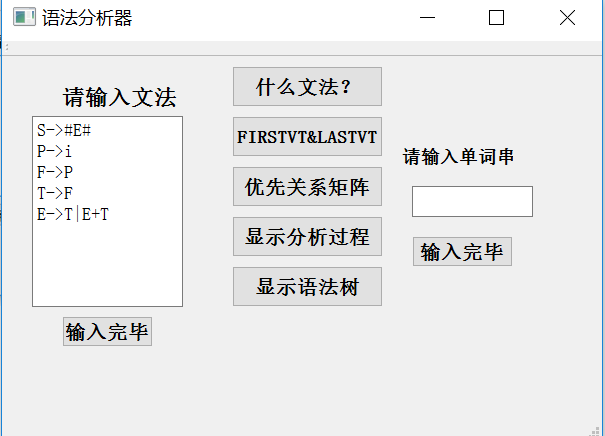
调试以及开发的过程：打开Qt,新建项目，选择主窗口基类是QMainwindow,然后设计ui，补充各部分代码，添加新的窗口。

运行结果：

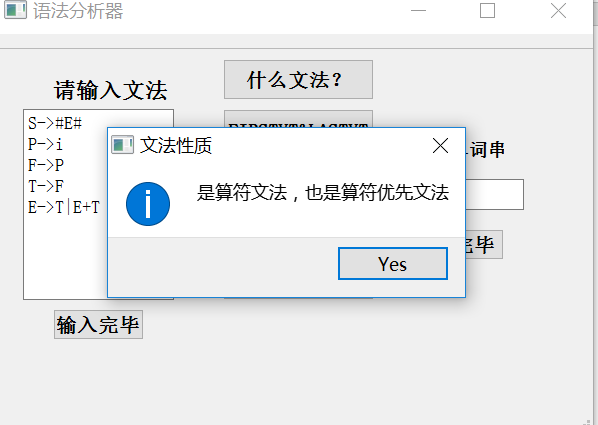
运行源程序，显示菜单。



1. 显示文法性质

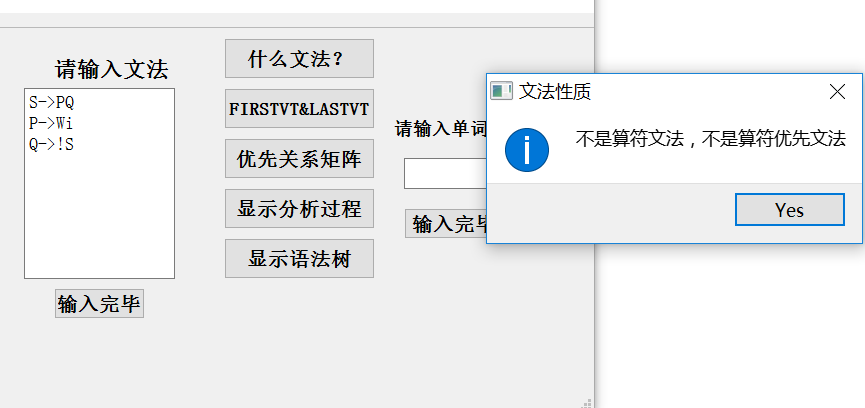


点击输入完毕，再点击“什么文法”，显示如下：



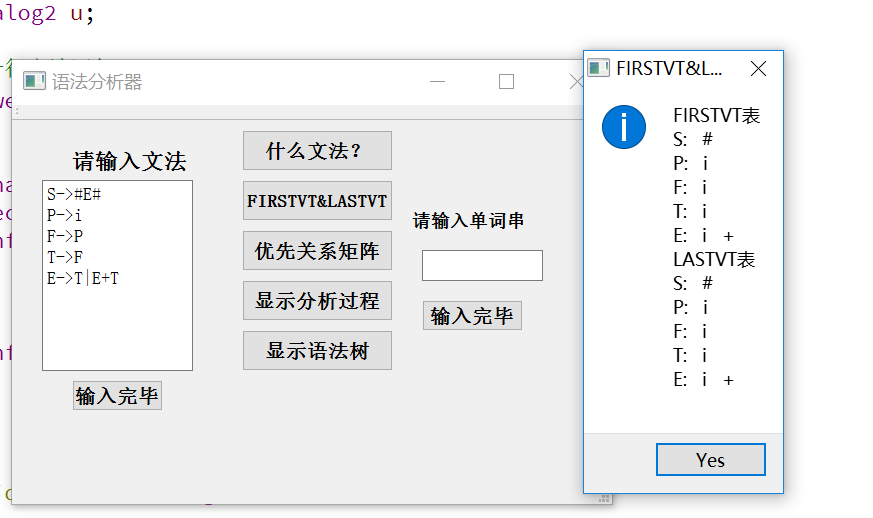
如果输入不对，会报错如下：

比如，输入的不是算符优先文法，会显示如下：



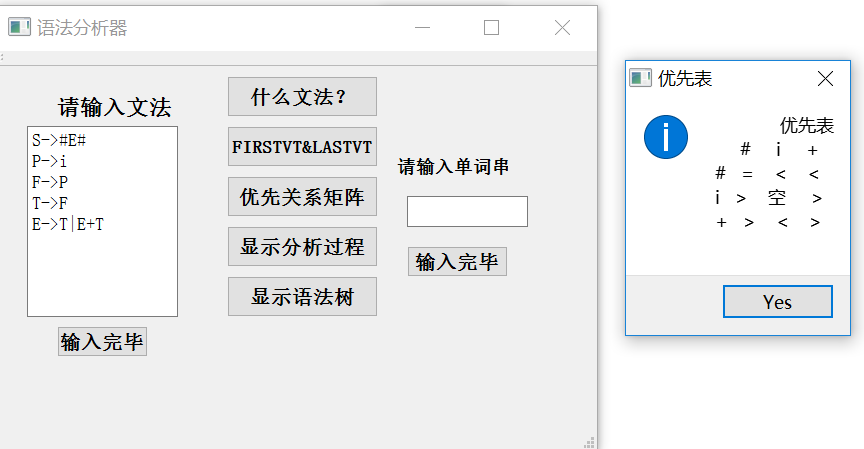
1. 显示FIRSTVT&&LASTVT

输入完毕点击后，点击“LASTVT&&FIRSTVT”



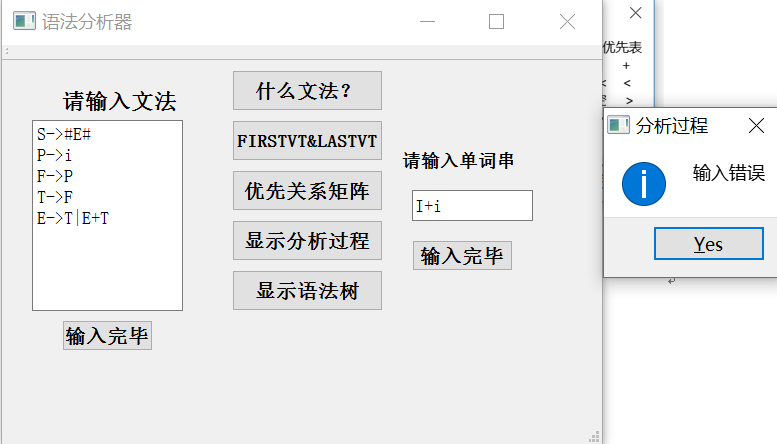
1. 显示优先关系矩阵

**必须在上一步显示FIRSTVT&&LASTVT后才能做这一步**



1. 输入单词串

当输入的不是合法（输入非终结符，警告纠错）

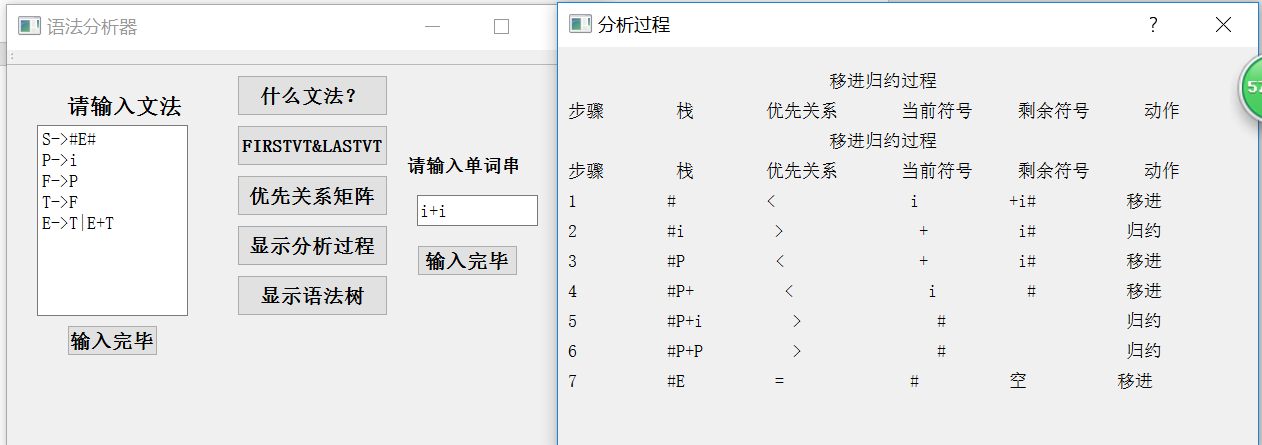


当输入合法是，点击输入完毕



1. 显示移进归约过程

点击“显示分析过程”



1. 显示语法树

点击“显示语法树”，得到一棵算符优先语法树

